Next Word Prediction Using RNN and Word Embedding

Pranay Javvaji
B-Tech, Computer Science and Engineering
Lovely Professional University
Phagwara, India
patelpranay577@gmail.com

Siddanth Gaikwad
B-Tech, Computer Science and Engineering
Lovely Professional University
Phagwara, India
siddgaikwad000@gmail.com

Enjula Uchoi
Lovely Professional University
Phagwara, India
enjulapaintoma@gmail.com

Abstract: Predicting the following word is also called text modeling. It is the work of projecting which word comes right away. It is one of the important part of NLP and has number of applications. I am working on building a model using the default Nietzsche text file that will guess the user's sentence after they type forty characters. The model will analyze those forty characters and suggest the likely top ten words. Using RNN neural organization which will be executed utilizing Tensor flow. Our aim is to train this model in such a way that it predicts 10 or more than 10 words as fast as possible taking less time. As RNN is LSTM it will get to know previous text and forecast the words which might be useful for the client to frame sentences and it uses letter prediction to the letter, in that it projects one letter after another to finally form a word.

***Index Terms*—Next Word Prediction, Language Modeling, NLP, RNN, TensorFlow,40 letters, Top 10 words, LSTM, Text prediction, Word prediction**

I. INTRODUCTION

A mathematical framework for language modeling can be viewed as a probabilistic distribution over a series of words. For a given sequence of length m, it associates a likelihood P(w1, w2, ..., wm) with the entire sequence. The model incorporates contextual information to differentiate between phrases and sentences that may appear alike. For instance in English, the phrases "recognize speech" and "wreck a pleasant beach" sound similar; but contain two very different things. Language Modelling (LM) perhaps lies at the very heart of problems in Natural Language Processing and Language Understanding. Models that might correctly position "distributions over sentences" capture not only the code complexities of language, such as grammatical structure but also distil a good amount of information regarding the knowledge that some corpora might contain.

This aspect is actually the largest part of AI that actually drives it, namely NLP. This lets people find better ways of communicating and learns from interaction with them- that one involves giving mobile users "predicted next words" that they type in line in applications, where the intention is to aid in the delivery of messages by having the client choose a suggested word instead of keying it in. As its long short time memory, LSTM would comprehend Past text and predict the words which may come in handy. This method aids in constructing sentences by forecasting text character by character, allowing individuals to compose words step by step. As creating essays or drafting extended paragraphs is often tedious, this strategy supports users in streamlining the task of text formation and facilitates seamless sentence and paragraph creation. Those include important parts of the paragraph and control the readers' attention. On the subject, not thinking about not staying idle, of what to write next. We would hope to build, at least simulate auto-complete functionalities using LSTM. Most software apply some of these techniques as well as To achieve this, NLP and normal neural networks do :- be trying this problem using LSTM by taking advantage The Default text file of Nietzsche or, in our case, training training data for the model

Traditional Neural Networks face challenges in managing large volumes of data. To address this issue, Recurrent Neural Networks (RNNs) were introduced, incorporating loops that allow them to retain information from prior inputs. Equipped with such loops, RNNs can always manage to recall context, working pretty well for historical context tasks.Consider this sentence: In Paris the climate is very unreliable; therefore, I always take my umbrella. Using the context provided by 'Paris' and the concept of 'Unrecognised Climate' maintained and regained by RNNs next word would be 'umbrella'. However for longer complex sentences or dependencies far removed RNNs face a lot of problems. Enter Long Short-Term Memory (LSTM), a special structure specifically designed for handling long-term dependencies. LSTMs are good at capturing context and thus suitable for such works. Additionally, our analysis showed that twu-directional LSTMs work better than one-directional ones. Therefore, in this paper, we use two-directional long-term memory. These networks learn on either ends of the input sequence both from right to left and from left side to right side simultaneously. This process of multi-training enhances the content of the word, therefore allowing swift combination within the greater Morphological model. Outcome: faster and more comprehensive learning enhances ability to solve problems.

Identify applicable funding agency here. If none, delete this.

II. Literature Review

It made the model even more niche as it only had the a Type-ahead prediction using the N-gram model. It is the Indo-Iranian language. They prepared the prototype in Kurdish text corpus. They have to endure tougher critics due to the indo-iranian textual language is almost negligible. This N-Gram saves typing time for the Kurdish. This model will detect the text : At the time of client input a text it asks user for inclusion of the following 5 phrases. The proposed system, as shown in the printed word or words above, will Suggest the following 5 terms. This Standard have an correctness of 96.03[1].

A Austroasiatic linguistic model applied an RNN. Traditional NN will just grasp terms they consists of and shown. Long-term dependencies are beyond the capabilities of the N-gram model. The model has more than twenty four M syllables constructed from fifteen hundred movie subtitles. In this paper, RNNs are It has been tested with a Vietnamese language model. Contributions That Summarized Below Follow: BuildingSyllable-based language model for the Vietnamese script with RNNs. constructing the Character-based text model for the Vietnamese language RNN-based model. Tested on a dataset of 24 million syllables consisting of fifteen hundred film captions subtitles. This prototype also summarizes that RNN-driven language system. achieves good outcomes. It achieved a Ambiguity score of 83.8 Based on the result [2], it seems to be a good model because it outperforms the N-gram model.

A paper on the Ukrainian tongue looks at the word prediction model, but it focuses on much more than that, specifically the Ukrainian dialect. The most important reason for a particular Ukrainian vernacular is that of They don't have too much for the Ukrainian vernacular. The serial direction completes the next word. succeeded. The Markov chains achieved the most accurate and fastest response times. The hybrid It turns out that the model is capable enough to give good results, though very slow. The objective of this paper is to explore current next-word prediction techniques based on the input text and evaluate them using Ukrainian language data.[3]

A. *Next Word Prediction*

Word prediction of words that have been submitted Algorithms give the user words from the possible list possibilities. Most of the Word prediction tools generate new phrases naturally as they are used. uses a user's previous input to foresee the future words daily vocabulary to forecast Phrases in the future. Despite word predictions being They would be used in improving typing speed and accuracy. Actually, it slows down the typing speed in some instances, especially When words are short. Guessing the word These technologies are for the good of people afflicted with difficulty in speaking as well as those who simply cannot writing speed. This research creates a major platform of speedy electronic communications for a language model. which, with some set of, tells forward what likely future words are Current phrases. Then, the word preceding that would be approximated by A predictive word algorithm that is thought to Continue with the following first few text pieces: It has also become common knowledge which word comes next. Also, the other term for Language, next-word predictions Modeling. NLP is involved in all these domains: with different contexts. Average accuracy of word predictions Unluckily, it only comes in consistency. about 30-40of the cases. Most of Smart phone keyboards have next word prediction. Besides, Google uses next word prediction based on how we use our browsers. As such, preloaded information is also Stored on our mobile phones' keyboard, which performs directly anticipate the next word. The next word predictions may help Students should be fluent in writing and then produce more outstanding writing. Audio accompaniment to authenticate word choices. Narrow the gap between what and why. Achievement through the ethos of the writing.

III. PROPOSED WORK

The work proposed in the paper appears to be challenging in predicting the next word with efficient output for a given sentence. It will occur, concentrating on aspects that enhance client live through by Live forecasts. For this we have - The need-be accounted for by typists while typing will develop a system which can predict the term succeeding exactly in relation to the input sequence, the word.

A. *DataSet*

The dataset is comments from two YouTube channels; these were combined for this study in order to make a single dataset. This data was used as the baseline for model to train and evaluate and it offers a well-balanced choice of texts that cover many themes and languagistic modes. The input will be processed with Bi-directional LSTM networks of the proposed model. Both forward as well as the reverse directions are followed for taking pictures of the preceding and succeeding contexts. It will make the model watchful of word relationships and their position in a sentence.Improvement of its predictive power with the next word.

B. *Preprocessing*

This pre-processing is very important because it eliminates unnecessary elements which would cause the model to degrade, and do not help to predict the next word. This cleaned data, as a preprocessing step, eliminates all meaningless terms which could mislead the experiment or degrade its precision. A Case study includes Graphic marks, Digits, Non-context terms, additionally hence, more noises could be filtered in the information. In our data, We have 10 columns and 6,508 entries, but the heading column must be used to forecast the following word . This area comprises news headings therefore it is It's short and informative too. Some unwanted characters and words have to be removed in the way they may influence. adversely affect the accuracy of the model. We can also omit the words like "the", "a", "an", etc. because these words do not play a big role in the semantics of a sentence. Next we use the Tokenization process, where we assign an Identifier for every term moreover create corresponding term count. Such a transformation reduces text into numerical values, thus Those words then could fell to a system. A term position is essentially the Lexicon that associates every term its respective identifier. Many papers aim at constructing a model that tries to predict the next text, but very few of them are verified for their effectiveness, such as in the work This one predicts what the next code will be using SVM N-gram and RNN. It is useful but not without its limitations with all the stated drawbacks that include the requirement of a large amount of data and overfitting. Now, in place, there can come up a new kind of algorithm like LSTM or Bidirectional LSTM, which would give better solutions For this issue, because they track lasting correlations and learn from both previous and upcoming contexts. The current system has some drawbacks which are followed by:

1. Challenges in Word Prediction Accuracy:

-The intricacy and ambiguity of human language make it intrinsically difficult to predict user-intended words.

- Because user preferences and writing styles differ, it might be tricky to predict the next word in a given context.

2. Traditional Algorithms' Drawbacks:

-SVM and Decision Tree algorithms don't do well in word prediction tasks.

- They don't take sentence structure and word order into account.

- They are not as scalable for big amounts of data because to their high memory and processing expenses.

3. Linguistic Adaptability and Model Maintenance:

-Neural networks need to be continuously trained on a variety of languages and changing data in order to improve predictions.

- As language changes, new terms and phrases are added.

- To keep up with linguistic changes, regular model upgrades with new vocabulary and data are crucial.

C. *Bi-directional LSTM vs LSTM*

Sequence prediction has always presented serious difficulties in the field of data science. Thanks to recent developments in data science, Long Short-Term Memory networks (LSTMs) have shown to be a very successful approach to these problems. Many different sequence prediction problems are handled by LSTMs. Think about how we arrange our daily schedules: we set out time for meetings, prioritize appointments, and even plan ahead for any changes. Standard Recurrent Neural Networks (RNNs) are not very effective in this area, though. On the other hand, LSTMs are excellent in quietly altering data via multiplicative and additive operations. They can convey important information because of their capacity to maintain context via cell states. In addition, we find Bi-directional LSTMs perform better than their unidirectional counterparts. Bidirectional LSTMs improve context comprehension and speed up learning by training on both ends of an intake order—processed both Succession as a result of left to right. This makes them an effective tool for sequence prediction tasks. In this sense, LSTMs are able to selectively remember or forget things.

D. *Contextual Adaptability and Selective Memory in LSTMs*

Though they are excellent at remembering context, LSTMs' real strength is in selective memory. Depending on the job at hand, these networks may selectively remember or forget information. Imagine a conversation where the topic of trip plans is discussed, followed by a recipe, and then a mention of a favorite book. The subject of the conversation may change quickly.

LSTMs adjust with ease, keeping vital context and eliminating unimportant elements. With two training processes, bi-directional LSTMs improve this flexibility. They get a comprehensive grasp of the sequence by taking into account both the past and the future environment. More precise predictions and quicker learning are made possible by this complex knowledge. In conclusion, LSTMs—both unidirectional and bidirectional—are essential tools for deciphering the complexities of language, allowing us to successfully predict and communicate.

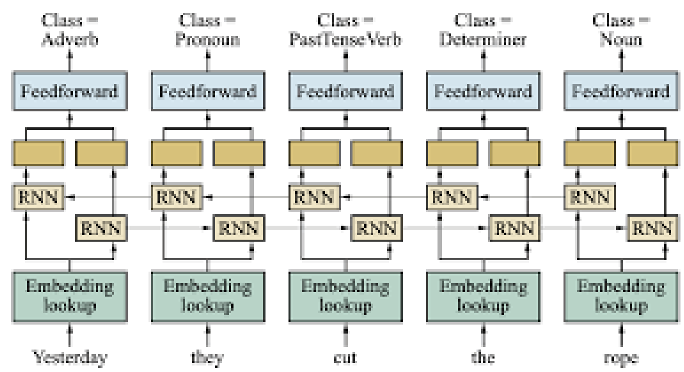

Fig. 1. RNN model for part-of-speech tagging.

The picture seems to show the architecture of a recurrent neural network (RNN) model used for some sort of natural language processing (NLP) job, such part-of-speech (POS) tagging or sequence labeling. After processing each word through recurrent and embedding layers, the model produces a label that corresponds to the grammatical role or class of each word. The model receives a series of words as input. Below is a detailed analysis of the image:

1. Enter Words:

- Sentence elements in the input include "Yesterday," "they," "cut," "the," and "rope." The order of these words is indicative of how sentences are processed in tasks using natural language.

- Every word is a component of a certain part of speech (POS). For instance, "cut" is a past-tense verb, "the" is a determiner, ”rope” is a noun, ”yesterday” is an adverb, and ”they” is a pronoun.

2. Lookup for embeddings:

- The embedding lookup is the initial layer in the model pipeline for every word. Words are usually represented in NLP tasks as dense vectors in a continuous space called embeddings. Semantic information and word associations are captured by these vectors.
- The words’ one-hot encoded or indexed representation is transformed into a continuous-valued vector by fetching the embeddings for each word from a pre-trained or learnt embedding matrix. Each word block’s bottom illustrates this, demonstrating how each word is processed through an embedding layer.

3. Recurrent Neural Network (RNN):

- The word representations are fed into a sequence of recurrent layers (represented as blocks labeled ”RNN”) after the embedding lookup. To handle sequential data, recurrent neural networks maintain a hidden state that retains information over different time steps.
- Word embeddings are processed one by one by each RNN block, taking into account the context that the words that came before it offered. This is essential to comprehending a sentence’s meaning and syntactic structure.
- Arrows in the diagram show how the concealed states are carried forward through the phrase, pointing from one RNN block to the next. This design captures the context-dependent character of language by enabling the model to take preceding words into account when predicting the current word.

4. Layers that feed forward:

- The feedforward layers (highlighted at the top of the model) come after the RNN layers and handle the hidden states created by the recurrent layers in more detail. Dense neural networks are usually utilized in these layers to convert the hidden representations into the desired outputs.
- The feedforward layers in NLP tasks such as POS tagging transfer the RNN’s hidden state into a classification decision, designating a particular grammatical class (e.g., Adverb, Pronoun, Noun, etc.) for each word.

5. Class Outcome:

- With each word’s anticipated class displayed at the top of the graphic. For instance, ”cut” is a past-tense verb, ”they” is a pronoun, and the term ”yesterday” is an adverb.
- After going through the feedforward, RNN, and embedding layers, each word is categorized individually; nevertheless, the RNN makes sure that the words have the same context.

6. Processing in Sequence:

- The whole architecture reads the text word by word, but because of the recurring connections, the class of the current word may be inferred from prior ones. One important characteristic of RNNs that sets them apart from models that handle inputs individually is their sequential nature.

IV. Results and Discussion :

A. *Comparative Evaluation and Model Effectiveness*

A comparison examination of many predictive models is presented in this part, emphasizing the higher results of the offered two-way LSTM (BI-LSTM) system . The following table provides a summary of the findings of the comparison, Which is determined by the accuracy of word prediction.

TABLE I
Table 1: Comparative Correctness of Predictions Model

Performance Analysis of Different Models		
Model Name	**No. of Words Predicted**	**Accuracy**
N-Gram Model [1]	The greatest repeatance or the top five key terms estimated depending on the frequency of word sequences.	92%
Long Short Term Memory (LSTM) [5]	It will tally 10 words and present the user with a list of them.	58.6%
BiDirectional LSTM (BILSTM) [Proposed Model]	It forecasts a set number of words according to the need.	93%

Examining the N-Gram models shown in Table 1, each of the five n-gram model types is applied, although the prototype just functions over a certain kind of Written dataset, that isn’t appropriate to many linguistics. The N-gram will be decreased when the machine cannot find enough evidence to predict the next word. Our model functions effectively.That never results in a reduction of accuracy.

The subsequent model is built on long short-term memory (LSTM), which has a poor level of accuracy overall. LSTM only stores data from the past since those are the only inputs it has ever seen. Long Short-Term Memory is outperformed by our model in Provisions of precision and data storage. The BI-LSTM model has a 93% accuracy rate, which is good. In contrast to a normal LSTM, a bidirectional LSTM allows input to flow in both ways. An input can only flow in one way with a traditional LSTM, either forward or backward.

With bidirectional input, we may have information flow in both directions while preserving the past and the future. For next word prediction, BI-LSTM turns out to be the most effective model as a result. This design has many practical applications, especially in natural language processing. The main cause of this is because each element of an input sequence consists of both current and historical data. As a result, Bi-LSTM can produce an output that is more relevant by combining LSTM layers from both directions.

B. Loss and Accuracy Visualization

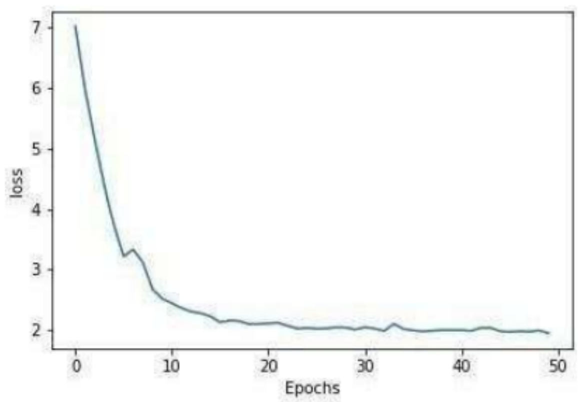

Fig. 2. **Loss**

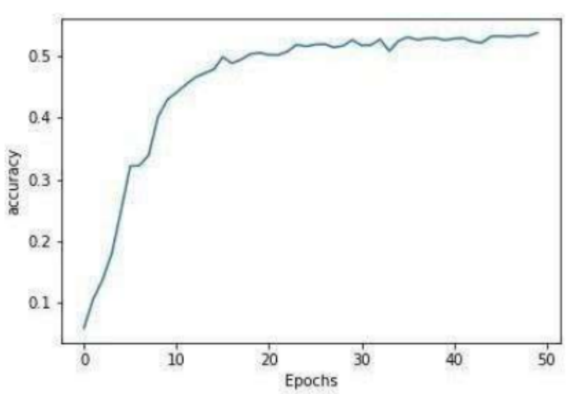

Fig. 3. **Accuracy**

C. SUMMARY AND FUTURE APPLICATIONS

In the following sentence, prediction is absolutely necessary both now and in the future. This approach is being tried by transitional companies since it makes them more approachable. Nevertheless, there is still a great deal of research to be done in this specific area. The bi-directional LSTM is employed to resolve the removed dependency problem as it utilizes memory cells to retain the unified sequence. The goal of this model is to design and evaluate an algorithm that achieves both exceptional accuracy and relevance for this application. This study illustrates how the system predicts and modifies the next or target words using various methods, and how TensorFlow improves the scalability of the developed model.

When something is presented in a different way, generally in a way that makes the original meaning clearer and is shorter and simpler, it is rephrased. Here, our computer will suggest more phrases that are related, making it simpler to create n statements that all have the same meaning. This method can assist end users in anticipating the next phrase because the prototype will be taught using a collection of music lyrics data in songs by creating melodies and lyrics, which is a significant area where this method might be beneficial.

By anticipating your next words as you type in the email body, Smart Compose builds on the capabilities of Smart Reply. Using an average word embedding in each field, this hybrid technique encodes the subject and previous email. Afterward, sum the Average vector representations and transmit Associate them with the target sequence during each decoding phase.

References

[1] S. Hochreiter and J. Schmidhuber, “LSTM can solve hard long-time lag problems,” in *Advances in Neural Information Processing Systems*, 1997, pp. 473-479.

[2] Y. Bengio, P. Simard, and P. Frasconi, “Learning long-term dependencies with gradient descent is difficult,” *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157-166, 1994.

[3] A. Rashid, H. K. Saeed, and T. A. Rashid, “Next word prediction based on the N-gram model for Kurdish Sorani and Kurmanji,” *Neural Computing and Applications*, vol. 33, pp. 4547–4566, 2021.

[4] A. F. Ganai and F. Khursheed, “Predicting next word using RNN and LSTM cells: Statistical language modeling,” in *2019 Fifth International Conference on Image Information Processing (ICIIP)*, 2019, pp. 469-474.

[5] S. Siami-Namini, N. Tavakoli, and A. S. Nami, “The performance of LSTM and BiLSTM in forecasting time series,” in *2019 IEEE International Conference on Big Data (Big Data)*, 2019, pp. 3285-3292.

[6] R. Sharma, N. Goel, N. Aggarwal, P. Kaur, and C. Prakash, “Next word prediction in Hindi using deep learning techniques,” in *2019 International Conference on Data Science and Engineering (ICDSE)*, 2019, pp. 55-60.

[7] K. Yeon, K. Min, J. Shin, M. Sunwoo, and M. Han, “Ego-vehicle speed prediction using a long short-term memory based recurrent neural network,” *International Journal of Automotive Technology*, vol. 20, pp. 713, 2019.

[8] M. Wang, W. Liu, and Y. Zhong, “Simple recurrent network for Chinese word prediction,” in *Proceedings of 1993 International Conference on Neural Networks (IJCNN-93)*, Nagoya, Japan, 1993, pp. 263-266.

[9] V. Raunak, V. Gupta, and F. Metze, “Effective dimensionality reduction for word embeddings,” in *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, Florence, Italy, 2019, pp. 235–243.

[10] Y. Bengio, P. Simard, and P. Frasconi, “Learning long-term dependencies with gradient descent is difficult,” *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157-166, 1994.

[11] A. Rashid, H. K. Saeed, and T. A. Rashid, “Next word prediction based on the N-gram model for Kurdish Sorani and Kurmanji,” *Neural Computing and Applications*, vol. 33, pp. 4547–4566, 2021.

[12] A. F. Ganai and F. Khursheed, “Predicting next word using RNN and LSTM cells: Statistical language modeling,” in *2019 Fifth International Conference on Image Information Processing (ICIIP)*, 2019, pp. 469-474.

[13] S. Siami-Namini, N. Tavakoli, and A. S. Nami, “The performance of LSTM and BiLSTM in forecasting time series,” in *2019 IEEE International Conference on Big Data (Big Data)*, 2019, pp. 3285-3292.

[14] R. Sharma, N. Goel, N. Aggarwal, P. Kaur, and C. Prakash, “Next word prediction in Hindi using deep learning techniques,” in *2019 International Conference on Data Science and Engineering (ICDSE)*, 2019, pp. 55-60.

[15] K. Yeon, K. Min, J. Shin, M. Sunwoo, and M. Han, “Ego-vehicle speed prediction using a long short-term memory based recurrent neural network,” *International Journal of Automotive Technology*, vol. 20, pp. 713, 2019.

[16] M. Wang, W. Liu, and Y. Zhong, “Simple recurrent network for Chinese word prediction,” in *Proceedings of 1993 International Conference on Neural Networks (IJCNN-93)*, Nagoya, Japan, 1993, pp. 263-266.